

<<计算机图形学>>

图书基本信息

书名：<<计算机图形学>>

13位ISBN编号：9787302186977

10位ISBN编号：7302186979

出版时间：2009-2

出版时间：清华大学出版社

作者：Francis S Hill, Jr.

页数：706

译者：胡事民,刘利刚

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<计算机图形学>>

前言

本书原著在国外是很有影响力的教材，被许多国外著名的大学所采用。译者在尽量保持原书精髓的基础上，对原书做了一些表述上的调整，以符合中文习惯，原书中的一些笔误，也给予了纠正。

本书译者中，胡事民、刘永进、张松海来自清华大学计算机科学与技术系，刘利刚来自浙江大学数学系，均在第一线从事计算机图形学的教学和科研工作。

在本书的翻译中，胡事民教授负责第1、2、8章，刘利刚副教授负责10、11、12章，刘永进副教授负责5、6、7章，张松海博士负责3、4、9章。

经过一年多的努力，本书终于完成翻译，并付印。

在翻译过程中，得到了许多同事和学生的帮助，研究生徐昆、高岳、李勇、张砚、雷励星、陈韬、张一飞、藏域、程明明、来煜坤、勒力、杨永亮、张国鑫、王王禹屏、边哲、佟强、白石、沈超慧、卢少平、陈中贵、陈仁杰、陈军、胡建伟、孟敏、张冬梅等参与翻译了部分初稿，在此一并表示感谢。

由于译者水平有限，本书翻译过程中难有疏漏和不妥之处，敬请读者批评和指正。

前言 对于有志于学习计算机图形学基本原理和技术，并且渴望自己能够编写优质图形学程序的读者，本书为他们提供了入门的详细介绍。

计算机图形学具有强大的生命力并正在持续发展。

动画电影已经展示给人们图形学知识所能做到的令人激动的视觉效果。

电脑游戏和因特网中的图形效果同样激励人们去学习如何自己动手做出这样的效果。

图形软硬件系统正在以惊人的速度朝着更好、更快和更廉价的方向发展。

每年从研究和业界中提出的大量新技术不断地涌现。

然而，图形学最基本的原理和方法，却保持着稳定和连贯，并且这些知识中的大部分可以通过一个图形学课程学习得到。

本书将图形学的一些基本思想和方法组织起来，通过阅读使读者从只会编写简单程序的水平，提高到能够自行设计并编写出优良图形学程序的水平。

适用的读者范围 该书被设定为本科高年级或研究生一年级的一个或两个学期的课程教材。

该书也可用于自学。

该书的主要对象是专业为计算机科学或计算机工程的学生，当然也适用于其他专业如物理和数学的学生。

对数学基础的要求 读者应该具备相当于大学一年级的数学水平；并具备初等代数、几何、三角函数和初等微积分的基本知识。

向量和矩阵的有关知识将有助于本书的阅读，但不是必须的。

本书在相关章节中介绍了向量和矩阵的基本知识，同时，附录中总结了向量和矩阵的关键概念。

计算机图形学中使用了大量的数学符号来表达线、面和视点之间的基本几何关系。

虽然每个数学符号都很容易理解，但所需的符号数量却令人生畏。

书中特别强调了使用每种符号或技术的原因，以及如何利用这些数学符号合理地描述图形程序中感兴趣的对象。

对编程水平的要求 一般情况下，读者应具有至少一个学期使用C、C++或Java编写程序的经验。

相当多有关图形的编程涉及到将几何关系直接转换为代码，包括使用变量、函数、数组、循环和条件测试等，而这些东西在不同编程语言中都是类似的。

本书使用的编程语言是C++，但是，有C语言背景的读者可以轻易地读懂大部分材料。

读者如果熟练掌握C中的结构或C++中的类，将非常有助于本书的阅读。

如用于描述复杂场景的图形对象结构，这些场景（如一个城堡或一架飞机）可能由许多部分组成，而这些部分本身又由复杂的更小部分组成。

如果能熟练使用链表或树结构也会有一些帮助，但不是必须的。

对于只懂得C但不懂得C++的读者来说，需要掌握面向对象的编程理念。

<<计算机图形学>>

本书中定义了许多有用的类（如网格、场景、相机和纹理等类），并且说明了这些类为什么方便易用。在某些场合下，为了使程序更加清晰易读，我们使用了面向对象编程中的一些技术，如继承和多态，不过，我们并不过分强调面向对象的方法。

计算机图形学 (OpenGL版) (第3版) 前言 教学理念 第3版与前两版相比，几乎被完全重新组织和改写，但是基本的教学理念没有改变：我们坚信计算机图形学是在实际使用中学习到的，一定要动手编写程序并进行测试，才能真正地掌握计算机图形学。

本书的一个主要目标就是让读者懂得如何将一个特定的设计任务转换成相应的几何结构，并找到一个合适的数学表达，最后将这个表达转换为算法和程序代码。

本书将以循序渐进的方式让读者首先学习到如何编写简单的程序来绘制简单的图像，然后一步一步深入地掌握如何生成更加复杂的特殊效果的技巧。

练习和问题 本书包含超过350道来源于实际问题的练习题。其中大部分都是让读者停下来进行思考但并不需要编程的那种类型，这可以让读者自己测试一下知识掌握的程度。

其余小部分则需要编程实现。

此外，在所有章节的末尾，列出了总共超过50个的实例分析。它们都是一些编程的项目，难易程度由简到繁，适合于作为课后作业。

这些案例分析对相应章节内的内容进行了扩展。

无论是否全部完成这些实例分析任务，它们都应该作为相应章节的一个有机组成部分。

每一个实例分析任务都注明了相应的难易程度，用以判断完成该任务所需的大致时间。

学生编程能力各异，编程过程不可预测，但可大致总结如下。

- 难易程度**
- I. 简单的练习，应该可以一晚完成。
 - II. 较难任务，应该可以一周内完成，读者需要一定的时间来设计程序，并且有足够的时间反复测试和调试程序。
 - III. 复杂任务，一般需要3周时间来完成设计和实现。

完成它可以被视为一项重大进步。

OpenGL的使用 刚开始学习计算机图形学时，一个很大的障碍就是如何绘制一幅图像。写一个程序通常很简单，但最终要通过某些工具在屏幕上绘制出来。

现在，这些工具已经存在并且很容易获取。

1992年，Silicon Graphics公司推出了OpenGL。

现在它已经成为最广泛使用的图形应用编程接口(API)之一。

OpenGL通过提供可以在程序中调用的方法接口来绘制图像，并且它可以在因特网上免费下载。

正如附录1中所述，在大专院校和工业界中使用的各种计算机系统上，都可以使用OpenGL。

它易于安装和使用，目前作为标准的图形API已经成熟。

它的制定和修改由OpenGL评审委员会(一个负责引导OpenGL发展的工业协会)负责。

OpenGL非常适用于计算机图形的一个特殊原因是因为它的设备独立性或可移植性。

不同院校的机房使用不同种类的计算机，学生可以利用OpenGL在任何一台计算机上开发并运行程序。

此外，该程序也可以在不同操作系统下、不同类型的另一台计算机上运行，两台计算机上生成的图像也一定相同。

OpenGL提供了丰富且易用的二维图形和图像操作的API函数，然而OpenGL最强大之处在于处理三维图形。

使用OpenGL，读者可以在一个学期之内学习并掌握制作精美动画效果的技巧。

OpenGL的发展历史 OpenGL的第一个版本1.0在1992年发布。

从那时起，OpenGL一直都是一个成功的跨平台的图形API。

第一个版本可以很好地兼容一个叫做IrisGL的专有API，这种API是由Silicon Graphics公司设计开发的。兼容的目的是为了建立一个工业标准。

<<计算机图形学>>

为此, Silicon Graphics公司联合其他几家图形硬件公司, 共同制定了一个开放标准, 命名为OpenGL。不足为奇, 在随后的10年, 人们努力使得OpenGL变得更好和更加强大。在这段时间, 图形硬件也发展迅猛, 产生了越来越强大的图形加速卡, 这使得程序开发者迫切需要一个能够达到OpenGL最佳效果的API。因此, OpenGL提供了一系列的扩展函数, 允许开发者在图形硬件新特征出现后就可以马上充分利用它们。

每当有一种强大的软件系统面世, 标准的问题也随即产生。对于OpenGL, 为使它朝着可控、有序的方向发展, Silicon Graphics成立了一个监督组织, 该组织被称作OpenGL 架构评审委员会(Architecture Review Board, ARB)。ARB的主要任务是指导OpenGL规范的制定和一致性测试。

OpenGL规范详尽地描述了每个OpenGL方法和标识符的使用方法。

目前, ARB的成员有3Dlabs、苹果公司、IBM、英特尔、nVidia、ATI、SGI和Sun Microsystems等。

OpenGL规范中定义的方法由各个图形硬件厂商在图形驱动程序中实现。这个过程需要大量的资金和时间, 所以, 通常在规范发布后过一段时间, 支持该规范的驱动程序才会发布。

最新的版本OpenGL 2.0在2004年9月发布, 它包含了强大的OpenGL着色语言(Shading Language), 将在第8章对此进行详细介绍。

它的前一版本是OpenGL 1.5, 该版本发布于2003年。

计算机图形学是一个发展迅速的领域, 这使得开发人员往往很急切地等待OpenGL官方版本的发布。

所以, 必须存在一种机制, 使得小规模增加的最新OpenGL功能可以马上在最新的图形硬件上开发和测试。

为此, ARB提供了一个明确定义的扩展机制。

在OpenGL扩展注册表(<http://oss.sgi.com/projects/projects/ogl-sample/registry>)中可以看到扩展的列表, 其中定义了命名习惯, 新扩展的指导方针和其他相关的扩展讨论。

访问扩展是一个平台相关的任务, 而可用的扩展取决于计算机上安装的图形硬件。

一些库如GLEW和GLEE可以用来简化在应用程序中使用扩展。

扩展机制使得OpenGL以一个合理的成本运行。

小规模改动就可以拥有最新硬件的强大功能。

这些扩展一旦被批准并在OpenGL扩展注册表中注册登记之后, 即可以迅速发挥作用。

当某个扩展已经成熟到可以加入OpenGL的核心功能中时, ARB就会讨论决定是否将其加入, 然后发布新版本的OpenGL。

这些内容将在第8章中进行讨论。

C++编程语言的使用 C++是大多数工程和计算机科学专业的首选编程语言。

相比于C, C++有几个明显的优点, 如引用传递函数参数, 这使得显式指针不再需要, 代码阅读更加易懂。

通过使用流, 文件的输入输出(I/O)也被极大地简化。

一般来说, C++中的所有I/O的语法都比C中更为清晰。

为了使得本书中的描述更为简单, 我们没有特别强调C++中运算符的使用。

此外, C++中的类很容易开发应用, 如二维或三维中的点、线、窗口或颜色, 这使代码更加简单而且更加鲁棒。

使用类之后, 具体的操作如几何对象细节隐藏、编写绘制函数包装绘制、测试某个对象是否相交等都更加清晰易读。

注重三维计算机图形学 由于PC上的游戏越来越流行, 电影中令人眼花缭乱的动画越来越多, 学生都对开发三维交互式的图形应用程序特别有兴趣。

为了能让读者更快地进入三维图形学的主题, 本书重新组织了第一版和第二版中的若干章节。

在很多地方, 二维和三维的概念结合在一起论述, 这可以帮助理解两者之间的相似和区别。

使用场景设计语言来描述三维场景 如果用原始的OpenGL命令来描述一个包含很多三维物体的场景，将会非常不雅观而且费时。例如，如果使用OpenGL的命令逐一地定义6个面来描述一个立方体将显得非常乏味。所以，在第5章中会介绍一个简单的场景描述语言SDL(在附录中有更加详尽定义)。使用这个描述语言，学生可以使用熟悉的词汇来描述场景，如“立方体”、“球”和“旋转”等，并建立包含这些词汇的文件。

这些文件可以在运行时读入到程序中。

本书的附录（或本书的网站）中给出了一个能够阅读SDL文件并建立文件中所描述物体的解释程序。这样，用OpenGL来绘制场景文件中的物体列表就变得非常简单。

本书的结构和课程安排 本书包含多于一个学期的教学内容，甚至多于两个学期的内容（对前两版也是这种情况）。

本书经过细致的编排，使得授课老师可以根据课程的长度和课程的背景选择不同的章节组合来授课。下面在介绍完各章的主要内容之后，将介绍几个建议的章节组合教学方案。

<<计算机图形学>>

内容概要

计算机图形学是利用计算机研究图形的表示、生成、处理和显示的一门重要的计算机学科分支，它是计算机科学中最活跃的分支之一。

近年来，随着计算机及互联网技术的迅速发展，计算机图形学正越来越深入我们的生活，它在工业建模、视频处理、游戏制作、影视特技、生物信息和医药医疗等各行各业都有着及其重要的作用。可以说，计算机图形学的应用无所不在，且日益广泛；已经成为计算机科学技术与其它应用学科之间沟通的桥梁，成为许多计算机从业人员的必备素质之一。

本书是一本国外很有影响的教材，为许多国外著名大学所采用。

本书通过最能代表技术发展状况的示例综合介绍了计算机图形学方面的原则和技巧，本书对每个概念都进行了详细介绍，阐述了其背后的数学原理，并给出了用OpenGL实现的代码以及实现结果展示。

新版本还为读者提供了计算机图形学领域的最前沿信息。

本书是计算机图形学课程的很好教材，也是计算机图形学专业人员的很好参考书。

<<计算机图形学>>

书籍目录

第1章 计算机图形学概述 1.1 什么是计算机图形学 1.2 计算机生成的图片用在哪里 1.2.1 艺术、娱乐和出版行业 1.2.2 计算机图形学、感知和图像处理 1.2.3 过程监视 1.2.4 仿真显示 1.2.5 计算机辅助设计 1.2.6 科学分析与体可视化 1.3 计算机图形学中制作图像的基本元素 1.3.1 折线 1.3.2 文本 1.3.3 填充区域 1.3.4 光栅图像 1.3.5 光栅图像的灰度和色彩表达 1.4 图形显示设备 1.4.1 线画显示 1.4.2 光栅显示器 1.4.3 视频卡/3D加速器 1.4.4 其他的光栅显示设备 1.4.5 硬拷贝光栅设备 1.5 图形输入的基本单元和设备 1.5.1 逻辑上的输入图形基元类型 1.5.2 物理输入设备的类型 本章小结 本章习题 进一步阅读第2章 OpenGL绘图入门 2.1 生成图像初步 2.1.1 设备无关的编程和OpenGL 2.1.2 窗口的编程 2.1.3 如何打开一个窗口画图 2.2 OpenGL的基本图形元素 2.2.1 几个点丛绘制的例子 2.3 OpenGL中的直线绘制 2.3.1 绘制折线和多边形 2.3.2 使用moveTo () 和lineTo () 绘制线段 2.3.3 绘制边校正的矩形 2.3.4 边校正矩形的长宽比 2.3.5 填充多边形 2.3.6 OpenGL中的其他图形元素 2.4 与鼠标和键盘的交互 2.4.1 用鼠标交互 2.4.2 键盘交互 2.5 程序中的菜单设计与使用 本章小结 案例分析 进一步阅读第3章 更多的绘图工具 3.1 概述 3.2 世界窗口和视口 3.2.1 窗口到视口的映射 3.3 裁减线 3.3.1 如何裁减一条线 3.3.2 Cohen-Sutherland裁减算法 3.4 正多边形、圆和圆弧 3.4.1 正多边形 3.4.2 正n边形的变种 3.4.3 绘制圆弧和圆 3.4.4 曲线的逐次细化 3.5 曲线的参数形式 3.5.1 曲线的参数形式 3.5.2 绘制参数曲线 3.5.3 极坐标形状 本章小结 案例分析 进一步阅读第4章 图形学中的向量工具第5章 物体变换第6章 使用多边形网格建模第7章 三维观察第8章 几何体的真实感渲染第9章 光栅显示工具第10章 曲线和曲面设计第11章 颜色理论第12章 光线跟踪基础附录1 图形工具：怎样获取和安装OpenGL附录2 计算机图形学的数学基础附录3 有用的类、例种以及SDL附录4 分形和曼德布洛特集合附录5 相对性和海龟绘图

章节摘录

第1章 计算机图形学概述 本章学习目标 纵览计算机图形学领域 描述重要的图形输入和输出设备 预览 1.1节 介绍计算机图形学的相关领域。

1.2节给出了一些当今计算机图形学应用的例子。

1.3节介绍用计算机生成图像的基本方法。

在1.3.4节中特别介绍了光栅图像的概念,它的应用贯穿本书始终。

1.4节描述了一些现在常用的图形显示设备,1.5节简要介绍用于交互式图形应用的各种输入设备。

本书原著在国外是很有影响力的教材,被许多国外著名的大学所采用。

译者在尽量保持原书精髓的基础上,对原书做了一些表述上的调整,以符合中文习惯,原书中的一些笔误,也给予了纠正。

本书译者中,胡事民、刘永进、张松海来自清华大学计算机科学与技术系,刘利刚来自浙江大学数学系,均在第一线从事计算机图形学的教学和科研工作。

在本书的翻译中,胡事民教授负责第1、2、8章,刘利刚副教授负责10、11、12章,刘永进副教授负责5、6、7章,张松海博士负责3、4、9章。

经过一年多的努力,本书终于完成翻译,并付印。

在翻译过程中,得到了许多同事和学生的帮助,研究生徐昆、高岳、李勇、张砚、雷励星、陈韬、张一飞、藏域、程明明、来煜坤、勒力、杨永亮、张国鑫、王王禹屏、边哲、佟强、白石、沈超慧、卢少平、陈中贵、陈仁杰、陈军、胡建伟、孟敏、张冬梅等参与翻译了部分初稿,在此一并表示感谢。

由于译者水平有限,本书翻译过程中难有疏漏和不妥之处,敬请读者批评和指正。

前言 对于有志于学习计算机图形学基本原理和技术,并且渴望自己能够编写优质图形学程序的读者,本书为他们提供了入门的详细介绍。

计算机图形学具有强大的生命力并正在持续发展。

动画电影已经展示给人们图形学知识所能做到的令人激动的视觉效果。

电脑游戏和因特网中的图形效果同样激励人们去学习如何自己动手做出这样的效果。

图形软硬件系统正在以惊人的速度朝着更好、更快和更廉价的方向发展。

每年从研究和业界中提出的大量新技术不断地涌现。

然而,图形学最基本的原理和方法,却保持着稳定和连贯,并且这些知识中的大部分可以通过一个图形学课程学习得到。

本书将图形学的一些基本思想和方法组织起来,通过阅读使读者从只会编写简单程序的水平,提高到能够自行设计并编写出优良图形学程序的水平。

适用的读者范围 该书被设定为本科高年级或研究生一年级的一个或两个学期的课程教材。

该书也可用于自学。

该书的主要对象是专业为计算机科学或计算机工程的学生,当然也适用于其他专业如物理和数学的学生。

对数学基础的要求 读者应该具备相当于大学一年级的数学水平;并具备初等代数、几何、三角函数和初等微积分的基本知识。

向量和矩阵的有关知识将有助于本书的阅读,但不是必须的。

本书在相关章节中介绍了向量和矩阵的基本知识,同时,附录中总结了向量和矩阵的关键概念。

计算机图形学中使用了大量的数学符号来表达线、面和视点之间的基本几何关系。

虽然每个数学符号都很容易理解,但所需的符号数量却令人生畏。

书中特别强调了使用每种符号或技术的原因,以及如何利用这些数学符号合理地描述图形程序中感兴趣的对象。

对编程水平的要求 一般情况下,读者应具有至少一个学期使用C、C++或Java编写程序的经验。

相当多有关图形的编程涉及到将几何关系直接转换为代码,包括使用变量、函数、数组、循环和条件测试等,而这些东西在不同编程语言中都是类似的。

<<计算机图形学>>

本书使用的编程语言是C++，但是，有C语言背景的读者可以轻易地读懂大部分材料。

读者如果熟练掌握C中的结构或C++中的类，将非常有助于本书的阅读。

如用于描述复杂场景的图形对象结构，这些场景（如一个城堡或一架飞机）可能由许多部分组成，而这些部分本身又由复杂的更小部分组成。

如果能熟练使用链表或树结构也会有一些帮助，但不是必须的。

对于只懂得C但不懂得C++的读者来说，需要掌握面向对象的编程理念。

本书中定义了许多有用的类（如网格、场景、相机和纹理等类），并且说明了这些类为什么方便易用。

在某些场合下，为了使程序更加清晰易读，我们使用了面向对象编程中的一些技术，如继承和多态，不过，我们并不过分强调面向对象的方法。

计算机图形学 (OpenGL版) (第3版) 前言 教学理念 第3版与前两版相比，几乎被完全重新组织和改写，但是基本的教学理念没有改变：我们坚信计算机图形学是在实际使用中学习到的，一定要动手编写程序并进行测试，才能真正地掌握计算机图形学。

本书的一个主要目标就是让读者懂得如何将一个特定的设计任务转换成相应的几何结构，并找到一个合适的数学表达，最后将这个表达转换为算法和程序代码。

本书将以循序渐进的方式让读者首先学习到如何编写简单的程序来绘制简单的图像，然后一步一步深入地掌握如何生成更加复杂的特殊效果的技巧。

练习和问题 本书包含超过350道来源于实际问题的练习题。

其中大部分都是让读者停下来进行思考但并不需要编程的那种类型，这可以让读者自己测试一下知识掌握的程度。

其余小部分则需要编程实现。

此外，在所有章节的末尾，列出了总共超过50个的实例分析。

它们都是一些编程的项目，难易程度由简到繁，适合于作为课后作业。

这些案例分析对相应章节内的内容进行了扩展。

无论是否全部完成这些实例分析任务，它们都应该作为相应章节的一个有机组成部分。

每一个实例分析任务都注明了相应的难易程度，用以判断完成该任务所需的大致时间。

学生编程能力各异，编程过程不可预测，但可大致总结如下。

难易程度 . 简单的练习，应该可以一晚完成。

. 较难任务，应该可以一周内完成，读者需要一定的时间来设计程序，并且有足够的时间反复测试和调试程序。

. 复杂任务，一般需要3周时间来完成设计和实现。

完成它可以被视为一项重大进步。

OpenGL的使用 刚开始学习计算机图形学时，一个很大的障碍就是如何绘制一幅图像。

写一个程序通常很简单，但最终要通过某些工具在屏幕上绘制出来。

现在，这些工具已经存在并且很容易获取。

1992年，Silicon Graphics 公司推出了OpenGL。

现在它已经成为最广泛使用的图形应用编程接口(API)之一。

OpenGL通过提供可以在程序中调用的方法接口来绘制图像，并且它可以在因特网上免费下载。

正如附录1中所述，在大专院校和工业界中使用的各种计算机系统上，都可以使用OpenGL。

它易于安装和使用，目前作为标准的图形API已经成熟。

它的制定和修改由OpenGL评审委员会(一个负责引导OpenGL发展的工业协会)负责。

OpenGL非常适用于计算机图形的一个特殊原因是因为它的设备独立性或可移植性。

不同院校的机房使用不同种类的计算机，学生可以利用OpenGL在任何一台计算机上开发并运行程序。

此外，该程序也可以在不同操作系统下、不同类型的另一台计算机上运行，两台计算机上生成的图像也一定相同。

OpenGL提供了丰富且易用的二维图形和图像操作的API函数，然而OpenGL最强大之处在于处理

<<计算机图形学>>

三维图形。

使用OpenGL，读者可以在一个学期之内学习并掌握制作精美动画效果的技巧。

OpenGL的发展历史 OpenGL的第一个版本1.0在1992年发布。

从那时起，OpenGL一直都是一个成功的跨平台的图形API。

第一个版本可以很好地兼容一个叫做IrisGL的专有API，这种API是由Silicon Graphics公司设计开发的。兼容的目的是为了建立一个工业标准。

为此，Silicon Graphics公司联合其他几家图形硬件公司，共同制定了一个开放标准，命名为OpenGL。不足为奇，在随后的10年，人们努力使得OpenGL变得更好和更加强大。

在这段时间，图形硬件也发展迅猛，产生了越来越强大的图形加速卡，这使得程序开发者迫切需要一个能够达到OpenGL最佳效果的API。

因此，OpenGL提供了一系列的扩展函数，允许开发者在图形硬件新特征出现后就可以马上充分利用它们。

每当有一种强大的软件系统面世，标准的问题也随即产生。

对于OpenGL，为使它朝着可控、有序的方向发展，Silicon Graphics成立了一个监督组织，该组织被称作OpenGL 架构评审委员会(Architecture Review Board, ARB)。ARB的主要任务是指导OpenGL规范的制定和一致性测试。

OpenGL规范详尽地描述了每个OpenGL方法和标识符的使用方法。

目前，ARB的成员有3Dlabs、苹果公司、IBM、英特尔、nVidia、ATI、SGI和Sun Microsystems等。

OpenGL规范中定义的方法由各个图形硬件厂商在图形驱动程序中实现。

这个过程需要大量的资金和时间，所以，通常在规范发布后过一段时间，支持该规范的驱动程序才会发布。

最新的版本OpenGL 2.0在2004年9月发布，它包含了强大的OpenGL着色语言(Shading Language)，将在第8章对此进行详细介绍。

它的前一版本是OpenGL 1.5，该版本发布于2003年。

计算机图形学是一个发展迅速的领域，这使得开发人员往往很急切地等待OpenGL官方版本的发布。

所以，必须存在一种机制，使得小规模增加的最新OpenGL功能可以马上在最新的图形硬件上开发和测试。

为此，ARB提供了一个明确定义的扩展机制。

在OpenGL扩展注册表(<http://oss.sgi.com/projects/projects/ogl-sample/registry>)中可以看到扩展的列表，其中定义了命名习惯，新扩展的指导方针和其他相关的扩展讨论。

访问扩展是一个平台相关的任务，而可用的扩展取决于计算机上安装的图形硬件。

一些库如GLEW和GLEE可以用来简化在应用程序中使用扩展。

扩展机制使得OpenGL以一个合理的成本运行。

小规模改动就可以拥有最新硬件的强大功能。

这些扩展一旦被批准并在OpenGL扩展注册表中注册登记之后，即可以迅速发挥作用。

当某个扩展已经成熟到可以加入OpenGL的核心功能中时，ARB就会讨论决定是否将其加入，然后发布新版本的OpenGL。

这些内容将在第8章中进行讨论。

C++编程语言的使用 C++是大多数工程和计算机科学专业的首选编程语言。

相比于C，C++有几个明显的优点，如引用传递函数参数，这使得显式指针不再需要，代码阅读更加易懂。

通过使用流，文件的输入输出(I/O)也被极大地简化。

一般来说，C++中的所有I/O的语法都比C中更为清晰。

为了使得本书中的描述更为简单，我们没有特别强调C++中运算符的使用。

此外，C++中的类很容易开发应用，如二维或三维中的点、线、窗口或颜色，这使代码更加简单而且更加鲁棒。

<<计算机图形学>>

使用类之后，具体的操作如几何对象细节隐藏、编写绘制函数包装绘制、测试某个对象是否相交等都更加清晰易读。

注重三维计算机图形学 由于PC上的游戏越来越流行，电影中令人眼花缭乱的动画越来越多，学生都对开发三维交互式的图形应用程序特别有兴趣。

为了能让读者更快地进入三维图形学的主题，本书重新组织了第一版和第二版中的若干章节。在很多地方，二维和三维的概念结合在一起论述，这可以帮助理解两者之间的相似和区别。

使用场景设计语言来描述三维场景 如果用原始的OpenGL命令来描述一个包含很多三维物体的场景，将会非常不雅观而且费时。

例如，如果使用OpenGL的命令逐一定义6个面来描述一个立方体将显得非常乏味。

所以，在第5章中会介绍一个简单的场景描述语言SDL(在附录中有更加详尽定义)。

使用这个描述语言，学生可以使用熟悉的词汇来描述场景，如“立方体”、“球”和“旋转”等，并建立包含这些词汇的文件。

这些文件可以在运行时读入到程序中。

本书的附录（或本书的网站）中给出了一个能够阅读SDL文件并建立文件中所描述物体的解释程序。

这样，用OpenGL来绘制场景文件中的物体列表就变得非常简单。

本书的结构和课程安排 本书包含多于一个学期的教学内容，甚至多于两个学期的内容（对前两版也是这种情况）。

本书经过细致的编排，使得授课老师可以根据课程的长度和课程的背景选择不同的章节组合来授课。

下面在介绍完各章的主要内容之后，将介绍几个建议的章节组合教学方案。

<<计算机图形学>>

编辑推荐

《计算机图形学(OpenGL版)第3版》配套的网站提供了书中使用到的代码、彩色插图、完整程序以及与《计算机图形学(OpenGL版)第3版》密切相关的参考素材。

以C++语言作为程序设计语言。

深入介绍3D图形学及其数学知识，使学生可以创建真实的3D图形。

每章后面都给出了实际的案例研究。

清晰地介绍了计算机图形学的原理和技术。

大量真实而逼真的示例。

大量实践练习（每章大概有30多道）。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>